Échantillonnage progressif guidé pour stabiliser la courbe d'apprentissage

François Portet, René Quiniou

Échantillonnage progressif guidé pour stabiliser la courbe d'apprentissage

Guided Progressive Sampling for Learning Curve Stabilization

François Portet[1] René Quiniou[2]

[1] Department of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, UK
[2] IRISA, INRIA, Université de Rennes 1, Campus de Beaulieu, Rennes, France

fportet@csd.abdn.ac.uk, quiniou@irisa.fr

Résumé

L'un des enjeux de l'apprentissage artificiel est de pouvoir fonctionner avec des volumes de données toujours plus grands. Bien qu'il soit généralement admis que plus un ensemble d'apprentissage est large et plus les résultats sont performants, il existe des limites à la masse d'informations qu'un algorithme d'apprentissage peut manipuler. Pour résoudre ce problème, nous proposons d'améliorer la méthode d'échantillonnage progressif en guidant la construction d'un ensemble d'apprentissage réduit à partir d'un large ensemble de données. L'apprentissage à partir de l'ensemble réduit doit conduire à des performances similaires à l'apprentissage effectué avec l'ensemble complet. Le guidage de l'échantillonnage s'appuie sur une connaissance a priori qui accélère la convergence de l'algorithme. Cette approche présente trois avantages : 1) l'ensemble d'apprentissage réduit est composé des cas les plus représentatifs de l'ensemble complet; 2) la courbe d'apprentissage est stabilisée; 3) la détection de convergence est accélérée. L'application de cette méthode à des données classiques et à des données provenant d'unités de soins intensifs révèle qu'il est possible de réduire de façon significative un ensemble d'apprentissage sans diminuer la performance de l'apprentissage.

Mots Clefs

Echantillonnage progressif, courbe d'apprentissage.

Abstract

Machine learning faces greater challenges with respect to efficiency and exactitude as the databases increase considerably in size. Although, large training sets are supposed to improve learning performance, the volume of data a learning algorithm can handle is limited. To tackle this problem, we propose to improve a progressive sampling method by guiding the construction of a small training set from large databases with minimal loss in exactitude. The guidance is performed by a priori knowledge about the data that speeds up the convergence of the algorithm. This brings three advantages: 1) the reduced dataset is composed of the most representative instances of the dataset; 2) the learning curve is stabilized; 3) the convergence detection is accelerated. The application of this method to classical data and to neonatal intensive care data shows that it is possible to reduce significantly a training set without decreasing the learning performance.

Keywords

Progressive sampling, learning curve.

1 Introduction

L'accumulation continue de gros volumes de données dans des domaines tels que les marchés boursiers, la vidéosurveillance, la médecine, etc. représente un vrai défi pour les projets de recherche. Bien que les techniques d'apprentissage artificiel soient les meilleures candidates pour analyser et découvrir automatiquement des relations dans les bases de données, le volume de données que chaque algorithme d'apprentissage peut manipuler est limité. En effet, la complexité des algorithmes d'induction les plus efficaces est en *O(n)*. Ainsi, le traitement des données d'unités de soins intensifs (USI), qui produisent de grands volumes de données (environ 1 Mo par patient par jour), peut être coûteux en temps et presque impossible à traiter.

Pour résoudre ce problème, deux approches s'opposent dans la littérature : partitionnement et réduction.

Le partitionnement des données consiste à diviser le jeu de données en sous-ensembles afin d'apprendre un ou plusieurs classificateur(s) (parfois de différents types) pour chaque sous-ensemble. Les sorties de ces classificateurs sont ensuite combinées pour former le réseau final de classification. Il s'agit plus généralement de l'apprentissage d' « ensembles de classificateurs » [1], tels que le bagging [2; 3] ou le boosting [4], ou toute combinaison de ces techniques [5]. Ces méthodes sont capables d'atteindre de très hautes performances cependant elles présentent quelques inconvénients. Plusieurs classificateurs pour chaque sous-ensemble de données doivent être calculés, ce qui augmente la consommation de ressources. De plus, l'interprétation humaine des modèles appris devient extrêmement difficile lorsque des classificateurs de différents types

sont combinés (par exemple, réseaux de neurones et arbres de décision).

Les techniques de réduction de l'ensemble d'apprentissage sont divisées en deux approches : la réduction des attributs et l'échantillonnage de l'ensemble d'apprentissage. La première approche consiste à transcrire l'information contenue dans de grands ensembles de données dans des dimensions plus petites en utilisant, par exemple, le filtrage, la discrétisation, l'analyse en composantes principales (ACP), etc. Cependant, ces méthodes nécessitent un prétraitement des données et modifient les exemples de l'ensemble d'apprentissage de telle manière, qu'il devient difficile d'interpréter le modèle appris. La deuxième approche, appelée *échantillonnage* consiste à chercher un sous-ensemble du large ensemble de données conduisant à un apprentissage satisfaisant. Les techniques de fenêtrage ou « windowing » et l'échantillonnage progressif sont des exemples de cette approche.

Une célèbre méthode de fenêtrage est présentée par Quinlan [6] dans le cadre de ID3. Dans cette méthode, un sous-ensemble d'exemples (une fenêtre) est choisi aléatoirement et utilisé pour apprendre une théorie qui est ensuite testée sur le reste des exemples. Si la qualité n'est pas suffisante, la méthode élargit la fenêtre en ajoutant des exemples mal classés, réapprend une théorie et ainsi de suite. Cependant, le fenêtrage peut conduire à une perte d'efficacité en temps de calcul. C'est pourquoi Fürnkranz [7] a proposé l'« integrative windowing » qui permet d'apprendre des théories sur un sous ensemble de données et de retirer les exemples couverts par les théories déjà induites. L'ensemble d'apprentissage diminue ainsi durant l'apprentissage d'où l'amélioration du temps de calcul. Certaines approches d'apprentissage en ligne, telles que l'apprentissage incrémental utilisant une mémorisation partielle [8], améliorent ce principe. Elles proposent une mise à jour, au fil du temps, de concepts appris en maintenant une base d'apprentissage et une base de concepts appris « passés » (base d'exemples et background). L'arrivée de nouveaux exemples permet de mettre à jour la base d'apprentissage en incluant les exemples « extrêmes » (sélection des exemples non couverts, abandon des exemples devenus inutiles). Ces techniques ont montré une forte amélioration de la consommation mémoire mais au détriment des performances d'apprentissage.

L'échantillonnage progressif (PS : progressive sampling) [9] a été proposé pour composer incrémentalement un ensemble d'apprentissage réduit à partir d'un vaste volume de données. Provost et collègues [9] et Breiman [10] ont montré que le sous-échantillonnage mène à de plus petits ensembles de données sans diminuer la qualité de l'apprentissage et sans modifier le format initial des exemples. Provost et collègues [9] ont également montré que PS est plus efficace que de traiter directement l'ensemble des données, car il conduit souvent à un apprentissage plus rapide et permet d'éviter le sur-apprentissage dans une certaine mesure. Ils ont défini une méthode pour détecter le sous-ensemble optimal à partir de la courbe d'apprentissage (exactitude – accuracy - en fonction de la taille). Le problème principal consiste à savoir jusqu'à quelle limite sous-échantillonner sans affecter l'exactitude [2]. Plusieurs recherches ont proposé l'adaptation de PS, par exemple : PSOS [11] pour traiter des ensembles de données non équilibrés ; NSC [12] pour raffiner la taille programmée du sous-ensemble ; pour l'estimation du point de convergence [13] ; pour définir le sous-ensemble initial [14] ; des adaptations spécifiques pour apprendre des règles d'association [15; 16]. Cependant, l'étude d'un sous-échantillonnage général plus efficace que le choix aléatoire n'a pas été beaucoup abordée [17].

Dans cet article, nous proposons une variante de PS qui utilise une heuristique pour guider le choix des exemples à ajouter à l'ensemble d'apprentissage incrémenté. Elle est basée sur une mesure de distance estimant à quel point un exemple a été « correctement » classé par le modèle appris. Cette mesure permet d'ajouter ensuite les plus mauvais exemples mal classés à l'ensemble d'apprentissage, guidant ainsi la composition de l'échantillonnage progressif jusqu'à la convergence sans perte d'exactitude (avec une exactitude parfois plus élevée). Cette méthode présente quatre avantages : 1) l'ensemble de données réduit se compose des exemples les « plus représentatifs » de l'ensemble de données ; 2) la courbe d'apprentissage est stabilisée ; 3) la détection de convergence est accélérée ; et 4) les déséquilibres observés sur le nombre d'exemples par classe peuvent être contrebalancés.

Ce document est organisé comme suit. PS est tout d'abord brièvement décrit en section 2. La section 3 détaille notre approche : l'échantillonnage progressif guidé (GPS – Guided PS). En section 4, les résultats d'expériences entreprises avec des ensembles de données classiques et provenant d'unité de soins intensifs sont détaillés. Enfin, la section 5 discute et conclut l'approche.

2 Échantillonnage Progressif (PS)

L'apprentissage supervisé consiste à fournir un ensemble d'apprentissage composé de N exemples décrits par L attributs $A_1 \times A_2 \times \ldots \times A_L$, ou A_l représente le domaine du l^e attribut, plus un attribut spécifique C, qui représente la classe de cet exemple (c'est-à-dire le concept à apprendre). L'ensemble des valeurs de sortie est considéré de cardinalité finie (classification et non régression). L'algorithme d'apprentissage exploite l'ensemble d'exemples et produit un modèle M qui sera utilisé pour prédire la classe de nouveaux exemples non classés (c'est-à-dire pour lesquels C est inconnu).

Dans le cas de PS [9], N est considéré comme étant très grand, de sorte que l'apprentissage direct à partir de l'ensemble d'apprentissage complet *FDS* (Full Data Set) est impossible. PS commence par créer un petit sous-ensemble de *FDS*, appelé *TS* (Training Set) et l'augmente de manière incrémentale jusqu'à ce que l'exactitude de l'apprentissage satisfasse un critère de convergence. Les

résultats attendus sont que *TS* soit plus petit que *FDS* et qu'il mène à une exactitude similaire. L'algorithme 1 montre les étapes générales de PS. Avant de commencer l'apprentissage, l'augmentation progressive de la taille de *TS* est planifiée. Puis, *TS* est utilisé pour apprendre le modèle *M* (arbre de décision, réseau de neurones, etc.) qui est testé jusqu'à ce que la convergence soit atteinte.

Soit *FDS* l'ensemble d'apprentissage complet
Soit $S = \{n_0, \ldots, n_K\}$ l'augmentation planifiée de la taille de l'ensemble d'apprentissage *TS*
Tant que *pas de convergence* **faire**
    TS ← calculerTS(*FDS*) // échantillonner n_k instances de *FDS*
    M ← apprendre(*TS*) // apprendre le modèle *M*
    évaluer(*M*,*FDS*) // tester le modèle appris sur *FDS*
Fin faire
Retourner *M*

Algorithme 1. Échantillonnage progressif.

2.1 Planification de la taille

La planification (scheduling) est le processus qui consiste à planifier les tailles successives de *TS*. Deux principales méthodes de planification ont été proposées. La planification arithmétique [18], $S(i)=n_0 + i.n_\delta$ (par exemple $S = \{10, 110, 210, \ldots, N\}$ pour $n_0=10$ et $n_\delta=100$) et la planification géométrique [9], $S(i)=a^i.n_0$ (par exemple $S = \{10, 20, 40, 80, \ldots, N\}$ pour $n_0=10$ et $a=2$). Provost et collègues [9] ont montré que la planification géométrique conduit à un échantillonnage asymptotiquement optimal pour la plupart des problèmes d'apprentissage. Cependant, définir la planification idéale est toujours un problème non résolu [12].

2.2 Sélection des exemples

Dans PS, le choix des exemples destinés à composer *TS* est effectué aléatoirement. Provost et collègues [9] pressentent qu'une sélection active pourrait améliorer les performances mais perturberait trop la courbe d'apprentissage et empêcherait alors une convergence correcte. Même si la prise en compte de connaissance *a priori* est reconnue comme un moyen d'améliorer l'apprentissage [4], sa contribution à PS n'a pas été suffisamment étudiée.

2.3 Apprentissage et évaluation du modèle

Une fois que *TS* est constitué, il est utilisé pour apprendre le modèle *M*. *M* est ensuite évalué sur l'ensemble *FDS*. La classification par *M* de chaque exemple est étiquetée correcte ou incorrecte (c'est-à-dire une erreur). L'exactitude (Accuracy) est calculée par la formule

$$Acc = correctes / (correctes + erreurs)$$

où *correctes* représente les classifications correctes et *erreurs* représente les classifications incorrectes du modèle *M*.

2.4 Détection de convergence

Le *TS* optimal est détecté au moyen d'une courbe d'apprentissage : c'est le point pour lequel l'amélioration de l'exactitude devient trop petite comparée à la taille croissante de *TS*. La courbe d'apprentissage représentée sur la figure 1 montre le rapport idéal entre l'exactitude du modèle (axe vertical) et la taille de *TS* (axe horizontal). Selon Provost et collègues [9], les courbes d'apprentissage ont typiquement une première partie à croissance rapide, une portion centrale qui croit plus lentement et enfin une partie en plateau à l'extrémité. n_{min} représente la taille de *TS* au début du plateau, pour lequel l'exactitude est proche de l'exactitude du modèle appris à partir de *FDS*. Ce point est défini comme étant le *point de convergence*.

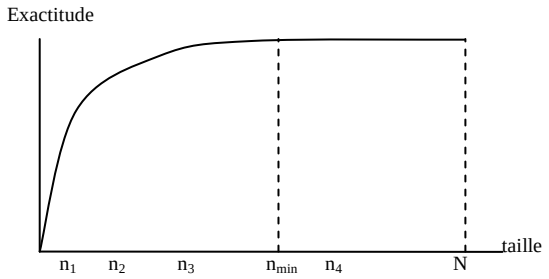

Fig. 1. Courbe d'apprentissage pour l'échantillonnage progressif.

Provost et collègues [9] supposent que dans la plupart des cas, la courbe est monotone croissante et définissent une méthode de détection de convergence appelée LRLS (régression linéaire avec échantillonnage local). Dans LRLS, pour chaque n_i courant, l sous-échantillonnages supplémentaires de taille voisine de n_i sont exécutés. Puis, une pente est estimée et si la pente tend vers 0, la convergence est détectée.
Bien que les auteurs aient rapporté une détection de convergence correcte, celle-ci nécessite des exécutions additionnelles. De plus, le postulat selon lequel la courbe est monotone croissante repose sur des études empiriques et contredit certaines études sur des bases de données statistiques [11; 19]. Comme nous le verrons par la suite, l'échantillonnage aléatoire peut mener à des courbes d'apprentissage qui ne croissent pas de manière monotone et pour lesquelles la convergence n'est pas atteinte.
Pour améliorer la stabilité de la courbe d'apprentissage, nous proposons une méthode d'échantillonnage progressif guidé, que nous appellerons GPS (Guided Progressive Sampling), qui utilise une mesure de distance pour guider le choix des exemples les plus appropriés à ajouter à *TS*.

3 Échantillonnage progressif guidé

Dans cette section nous présentons la méthode d'échantillonnage progressif guidé. Nous commençons

par décrire l'algorithme GPS, puis nous détaillons les diverses opérations de l'algorithme : composition de l'ensemble d'apprentissage initial, évaluation du modèle appris, mise à jour de l'ensemble d'apprentissage et détection de la convergence.

3.1 Algorithme GPS

GPS incrémente progressivement l'ensemble d'apprentissage en choisissant les exemples susceptibles de faire évoluer le plus rapidement possible l'apprentissage vers le modèle idéal. Pour ce faire, GPS utilise une mesure de distance *d* qui estime le degré de « mauvaise » classification d'un exemple. Le fait de choisir les exemples mal classés les plus éloignés comme les plus « appropriés » à ajouter à *TS* doit permettre de corriger le modèle *M* au plus tôt et ainsi accélérer la convergence de l'apprentissage. L'algorithme 2 présente l'introduction de *d* dans l'algorithme 1.

Soit *FDS* l'ensemble d'apprentissage complet
Soit *TS* l'ensemble d'apprentissage courant
Soit $S = \{S_1, S_2, S_3, \ldots, S_{NC}\}$, l'ensemble des incréments planifiés pour chaque classe
Soit *d* la mesure de distance d'un exemple à sa classe
Tant que *pas de convergence* **faire**
TS ← calculerTS(*R*, *FDS*, *S*) // sélectionner les meilleures instances de *FDS* selon *S* et *R*
M ← apprendre(*TS*) // apprendre le modèle *M* à partir de *TS*
R ← évaluer(*M*, *FDS*, *d*) // tester le modèle *M* appris sur *FDS*: calculer le score de chaque instance avec *d*
Fin faire
Retourner *M*

Algorithme 2. Échantillonnage progressif guidé.

L'algorithme utilise *FDS* pour construire *TS*. La manière dont *TS* évolue est contrôlée par le plan d'incrémentation *S*. Si *NC* classes sont à apprendre, *S* consiste en un ensemble de *NC* plans spécifiques à chaque classe. Ainsi, l'apprentissage peut choisir d'augmenter les classes minoritaires plus rapidement que les classes majoritaires afin d'accélérer une convergence au cours des premières itérations quand le nombre d'exemples dans chaque classe est plus équilibré.

3.2 Composition du *TS* initial

La composition du *TS* initial est réalisée en choisissant le nombre d'exemples exigés pour chaque classe. Différentes méthodes peuvent être employées. Gu et collègues [14] ont proposé la méthode SOSS (Statistical Optimal Sample Size) pour définir la dimension initiale de *TS*. SOSS consiste à déterminer un ensemble de petite taille qui ressemble suffisamment à l'ensemble de données entier. Dans, Cebron et Berthold [20] les auteurs utilisent la méthode de *Fuzzy c-means clustering* pour partitionner *FDS* et choisissent ensuite les centres des clusters comme *TS* courant. Dans notre cas, par une approche similaire, le *TS* initial est construit en 3 phases :

1. les centres de gravité des exemples $e_i \in FDS$ de chaque classe dans *FDS* sont calculés ;
2. pour chaque exemple $e_i \in FDS$, la distance $d(e_i)$ entre e_i et le centre de gravité de sa classe est calculée ;
3. les $e_i \in FDS$ dont la $d(e_i)$ est la plus grande sont retenus pour composer le TS initial.

La distance *d* ne donne qu'une estimation du degré d'incorrection de la classification de e_i par *M*. La distance *d* peut être plus ou moins précise (à noter que si elle était très précise elle devrait être utilisée pour l'apprentissage !). Plusieurs mesures de distance peuvent être utilisées, par exemple la distance euclidienne entre e_i et le centre de gravité des exemples de sa classe dans *FDS*.

3.3 Évaluation du modèle

Chaque exemple $e_i \in FDS$ est examiné pour former le triplet $(e_i, m(e_i), d(e_i))$ où $m(e_i) \in \{correct, erreur\}$ est le résultat de la classification de e_i en utilisant le modèle *M*, et $d(e_i)$ est la distance entre e_i et le centre de gravité de sa classe calculée lors de l'initialisation.

3.4 Mise à jour de *TS*

Après l'évaluation du modèle courant, *TS* est incrémenté en ajoutant à chaque classe, une partie des plus mauvais exemples mal classés de *FDS* qui ne sont pas déjà dans *TS*. Ils correspondent aux erreurs ayant les scores $d(e_i)$ les plus élevés. S'il n'y a plus de classifications incorrectes, les plus mauvaises classifications correctes sont ajoutées. Aucun exemple n'est ajouté deux fois (pas de sur-échantillonnage). Cette stratégie améliore la robustesse de l'apprentissage en attribuant plus de poids aux cas difficiles et repose sur l'hypothèse suivante. L'apprentissage consiste à trouver, dans l'espace des exemples, les frontières qui définissent le groupe d'exemples qui appartiennent à une même classe. Pendant l'échantillonnage progressif, les frontières apprises évoluent selon les exemples ajoutés dans *TS*. Ajouter uniquement les exemples mal classés, comme dans le fenêtrage [7], permet d'étendre les frontières.
Cependant, le fait d'ajouter ces seuls exemples incorrects peut également donner trop d'importance aux artefacts (exemples trop près des frontières de plusieurs classes qui ne sont pas intéressants à considérer ou exemples qui pourraient correspondre à du bruit). Ainsi, pour améliorer la stabilité de l'apprentissage, des exemples correctement classés sont également considérés dans une autre version de GPS (appelé GPS+) pour compenser l'ajout des exemples trop près des frontières.
Dans GPS+, les meilleurs exemples classés corrects à ajouter aux TS sont ceux ayant les valeurs $d(e_i) \in FDS$ les plus basses qui ne sont pas déjà dans *TS*. Ainsi GPS+ considère tous les exemples extrêmes (les meilleures classifications correctes et les plus mauvaises

classifications incorrectes). Les classifications incorrectes étendent les frontières et améliorent ainsi la sensibilité et les classifications correctes augmentent la stabilité de l'apprentissage et améliorent ainsi la précision. Fürnkranz [7] a montré que c'est une bonne stratégie en cas d'ensembles de données bruitées.

La complexité de l'échantillonnage est liée à la mesure de distance. Cependant, en pratique, pour une mesure euclidienne, l'échantillonnage guidé ne prend pas plus de temps que l'échantillonnage aléatoire, car le centre de gravité de chaque classe ainsi que la distance entre chaque e_i et le centre de gravité sont calculés seulement une fois à l'initialisation de l'algorithme. C'est une différence notable avec les algorithmes dont les centres de gravité évoluent [20] : ils doivent recalculer toutes les distances et donc, si *d* était en *O(n)*, la procédure d'évaluation « évaluer(*M, FDS, d*) » serait en $O(n^2)$. Or dans notre cas, la complexité reste en *O(n)*. Enfin, un échantillonnage aléatoire réellement représentatif, peut également s'avérer difficile [21].

3.5 Détection de la convergence

La détection de convergence est toujours un problème non résolu dans l'échantillonnage progressif. La méthode de convergence LRLS [9] suppose que la courbe d'apprentissage augmente de manière monotone. Comme nous le verrons par la suite, cette hypothèse n'est pas admissible pour notre méthode et les ensembles de données utilisés. Plutôt que d'utiliser LRLS, dont les exécutions additionnelles consomment trop de temps de calcul, nous avons adopté une approche semblable à celle d'Estrada et Morales [12], qui utilisent les points de la courbe d'apprentissage déjà calculés pour calculer une régression linéaire sur les derniers points de la courbe et détecter la convergence lorsque la pente tend vers 0.

4 Expérimentations

La méthode proposée a fait l'objet de tests empiriques sur plusieurs bases de données. Trois bases, de taille petite et moyenne, proviennent du « UCI repository » [22]. La dernière base contient des données enregistrées en unité de soins intensifs. Elle est particulièrement intéressante car elle est très grande, très déséquilibrée (c'est-à-dire que les exemples ne sont pas uniformément distribués parmi les classes) et présente du bruit. Pour chaque base de données, trois expériences ont été effectuées : échantillonnage progressif aléatoire (RS - random sampling), GPS considérant les classifications incorrectes seulement, et GPS+ considérant les classification correctes et incorrectes. Après la description de l'expérimentation, les résultats globaux sont décrits et les courbes d'apprentissage obtenues pour chaque expérience sont comparées.

4.1 Description de l'expérimentation

Le tableau 1 décrit les bases de données considérées pour les expériences. Trois bases de données proviennent du « UCI repository » [22] : « iris » qui fournit des attributs décrivant 3 classes de fleurs, « segment » qui se compose d'attributs décrivant une image d'extérieur à classifier (classe : herbe, ciel...), et « lettre » où chaque exemple doit être classé comme une lettre de l'alphabet.

Base	Taille	N	Atts.	Classes	n_0	n_δ
Iris	8KB	150	4	3	15	6
Segment	108KB	810	19	7	81	14
Lettre	687KB	20000	17	26	200	520
Bradycardie (#16234)	20MB	80408	25	2	{533,798}	{0,2680}

Table 1. Bases de données utilisées pour l'apprentissage.

La base de données de bradycardies vient de l'étude de Quinn et de Williams [23] et contient treize séries chronologiques de 24 heures enregistrant le rythme cardiaque de bébés prématurés recevant des soins intensifs. Les épisodes de bradycardie ont été annotés par deux experts cliniciens. Contrairement aux bases précédentes, cet ensemble de données est très déséquilibré. En effet, pour l'enregistrement #16234 la classe *bradycardie* se compose de 533 exemples tandis que la classe *non bradycardie* se compose de 79875 exemples. Ainsi les exemples de bradycardie représentent seulement 0,66% de l'ensemble de données (*FDS*). Cependant, c'est un cas normal car les bradycardies sont généralement des événements momentanés. En outre, les enregistrements contiennent des épisodes de bruit qui peuvent perturber l'apprentissage.

Pour chaque base de données, un plan d'incrémentation arithmétique a été généré en utilisant les paramètres donnés dans le tableau 1. Nous avons choisi un plan arithmétique pour l'étude car, malgré sa plus grande consommation en temps, il permet une acquisition plus précise de la courbe d'apprentissage que le plan géométrique. Pour les trois bases UCI, la taille de départ a été empiriquement définie comme étant 10% de la taille globale. Pour la base de données de bradycardie, les paramètres du programme ont été précisés spécifiquement pour chaque classe. Pour l'enregistrement #16234, le plan de la classe *bradycardie* est *533* tandis que le plan de la classe *non bradycardie* est *798+(i-1)*2680* ; 533 est le nombre total d'exemples de *bradycardie*, 2680 représente 3% de l'ensemble des exemples de *non bradycardie* et *i* est la i^e itération. Ce choix réside sur le fait que tous les exemples de la classe bradycardie sont *a priori* intéressants étant donné leur nombre très restreint. Pour la classe *non bradycardie*, le plan $S(i)=n_0 + i.n_\delta$ (cf. section 2.1) démarre avec un n_0 très proche du nombre d'exemples de la classe *bradycardie* pour tenter d'équilibrer *TS* au départ. L'accroissement n_δ a été choisi de manière empirique. RS utilise le même plan d'incrémentation et la même méthode de convergence que GPS et GPS+ mais les exemples à ajouter sont choisis de manière aléatoire.

Le modèle appris est un arbre de décision, produit par C4.5 avec élagage. Comme tous les attributs sont

numériques, la distance *d* choisie est la distance euclidienne entre un $e_i \in FDS$ et le centre de gravité de la classe à laquelle appartient e_i.

4.2 Résultats globaux

Le tableau 2 montre les résultats globaux, des méthodes RS, GPS, et GPS+. La colonne *FDS* donne l'exactitude (Acc.) des résultats d'apprentissage en utilisant *FDS* et la taille initiale de *FDS* en nombre d'exemples. Les autres colonnes donnent, pour chaque méthode, les résultats d'apprentissage, l'itération (iter.) à laquelle la méthode a convergé ainsi que la taille optimale trouvée en pourcentage de la taille de *FDS*. Pour les petites bases de données (iris et segment) GPS et GPS+ sont clairement supérieurs à RS. La convergence est plus rapide et l'exactitude est légèrement meilleure. RS ne converge pas sur la base de données « lettre ». Ce comportement a été également rapporté dans d'autres études [11]. Pour l'enregistrement #16234 de la base de données de bradycardies, GPS converge légèrement plus vite que RS, avec une meilleure exactitude. Par ailleurs, GPS présente une distribution des classes de (8%, 92%) en fin d'apprentissage tandis que le choix aléatoire donne (0,58%, 99%). Ainsi GPS a contribué à compenser la distribution non équilibrée.

	FDS		**RS**			**GPS+**			**GPS**		
Base	Acc.	taille	Acc	iter.	taille	Acc	iter.	taille	Acc	iter.	taille
Iris	98,0	150	98,0	18	80%	98,0	10	48%	98,0	10	48%
Segment	98,8	810	98,0	44	86%	99,0	37	75%	99,0	37	75%
Letter	96,4	20000	-	-	-	96,4	25	72%	96,3	22	65%
Bradycardies (#16234)	99,9	80408	99,5	4	11%	99,7	3	8%	99,7	3	8%

Table 2. Résultats des expérimentations.

Les résultats globaux montrent que GPS est supérieur à RS (et parfois est même supérieur à l'apprentissage direct avec *FDS*). GPS est également légèrement supérieur à GPS+ pour lequel l'addition des exemples positifs n'a pas amélioré l'exactitude des résultats et retarde ainsi la convergence. Ceci confirme le principe général selon lequel l'apprentissage est davantage amélioré par les exemples extrêmes. Les sections suivantes montrent que GPS est non seulement supérieur à RS en terme d'exactitude mais qu'il stabilise également la courbe d'apprentissage.

4.3 Courbes des apprentissages expérimentaux

Les courbes d'apprentissage obtenues avec les bases de données de l'UCI sont montrées Figure 2. Pour chaque base, les courbes d'apprentissage obtenues avec RS, GPS et GPS+ sont tracées. Le point de convergence est indiqué par un grand cercle pour GPS+, une étoile pour GPS et un losange pour RS.

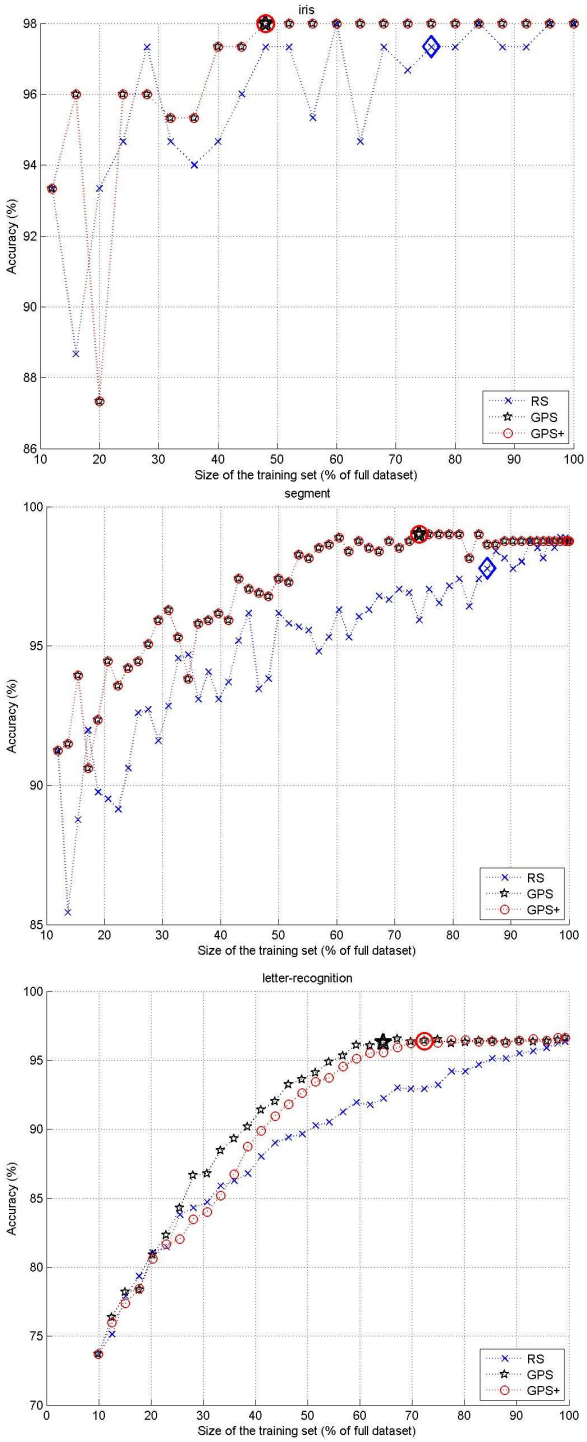

Fig. 2. Courbes d'apprentissage. Haut: iris; milieu: segments; bas: lettre.

Les différentes courbes montrent clairement l'intérêt de guider le choix des exemples par une mesure de distance. GPS converge plus rapidement que RS et présente également une courbe d'apprentissage plus stable.Les courbes de GPS sont en effet beaucoup plus proches de la description de la courbe d'apprentissage idéale donnée dans la section 2.4 que le choix aléatoire. L'amélioration principale est réalisée au niveau du plateau. GPS atteint

toujours un plateau qui correspond parfois à une exactitude plus élevée qu'un apprentissage sur *FDS* (cf. segments), tandis que le plateau de RS n'est pas stable. La pente est aussi plus raide avec GPS. Ainsi GPS démontre non seulement une convergence plus rapide mais également une meilleure exactitude.

4.4 Bradycardies

Les résultats obtenus avec les 13 enregistrements de la base de données de bradycardies sont présentés tableau 3. Pour chaque enregistrement, RS a été exécuté 20 fois et des tests de significativité ont été réalisés pour l'exactitude et l'itération de convergence.

D'une manière générale, et contrairement aux expériences précédentes, RS converge avec en moyenne une itération de moins que GPS+ et GPS. Selon les enregistrements, ce comportement n'est pas toujours observé et est rarement significatif (régulièrement p>0,05). Il peut donc être attribué à certaines particularités des données. Par contre l'exactitude de GPS+ et GPS est toujours supérieure à RS et cette différence est significative dans tous les enregistrements (le moins significatif étant p<0.04).

La figure 3 présente la courbe obtenue avec l'enregistrement #16234. Dans ce cas, GPS converge plus vite que RS avec une meilleure exactitude : Acc=99,7% contre Acc=99,5%. GPS atteint un plateau à la troisième itération qui reste à la même exactitude optimale jusqu'à la fin, alors que l'exactitude de RS continue à augmenter jusqu'à la dernière itération tout en restant inférieure à celle de GPS. Ceci suggère que tous les exemples ajoutés de la 3^{e} itération jusqu'à la dernière itération n'ajoutent rien à l'apprentissage pour GPS.

Ainsi 8% de *FDS* est suffisant pour atteindre une bonne exactitude. De plus, les exemples de bradycardie constitue 8% de *TS* avec GPS tandis qu'il représente 0,58% de *TS* avec RS. Ceci montre l'avantage de définir un plan séparé pour chaque classe en cas d'ensembles de données déséquilibrés.

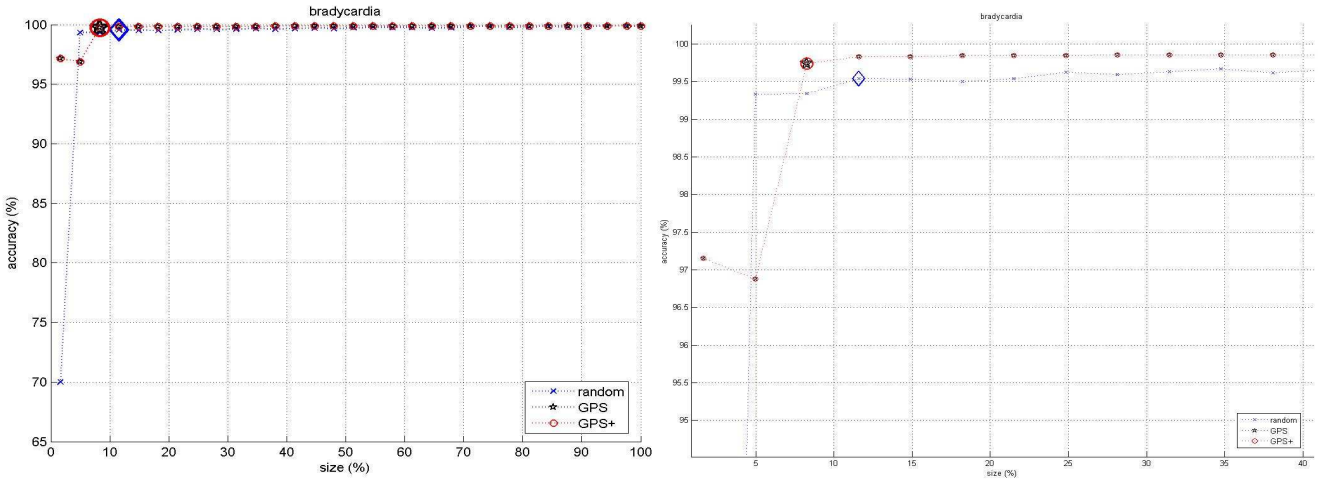

Fig. 3. Courbe d'apprentissage sur la base de données de bradycardies (gauche) et zoom sur le plateau (droite).

Enregistrement	RS acc.	RS it.	GPS+ acc.	GPS+ it	GPS acc.	GPS it
16227	$99,78^{+-0,02}$	$6,6^{+-0,8}$	99,90 (p<0,0001)	6 p<0,44)	99,90 (p<0,0001)	6 (p<0,44)
16228	$99,67^{+-0,02}$	$5,6^{+-1,0}$	99,83 (p<0,0001)	7 (p<0,19)	99,86 (p<0,0001)	7 (p<0,19)
16229	$99,10^{+-0,14}$	$8,0^{+-2,1}$	99,52 (p<0,01)	10 (p<0,36)	99,54 (p<0,01)	8 (p=1)
16230	$99,64^{+-0,06}$	$6,8^{+-1,3}$	99,86 (p<0,001)	8 (p<0,37)	99,87 (p<0,0001)	7 (p<0,92)
16231	$99,91^{+-0,03}$	$6,5^{+-0,7}$	99,97 (p<0,04)	6 (p<0,43)	99,97 (p<0,04)	7 (p<0,52)
16232	$99,55^{+-0,03}$	$6,0^{+-0,9}$	99,76 (p<0,0001)	9 (p<0,001)	99,75 (p<0,0001)	8 (p<0,03)
16233	$99,92^{+-0,02}$	$5,0^{+-0,0}$	99,99 (p<0,01)	6 (p=0)	99,98 (p<0,01)	5 (p=1)
16234	$99,60^{+-0,06}$	$7,0^{+-1,3}$	99,86 (p<0,0001)	7 (p<0,98)	99,86 (p<0,0001)	6 (p<0,48)
16235	$99,76^{+-0,04}$	$6,5^{+-0,6}$	99,91 (p<0,0001)	8 (p<0,02)	99,92 (p<0,0001)	8 (p<0,02)
16236	$99,76^{+-0,02}$	$6,4^{+-1,0}$	99,92 (p=0)	9 (p<0,01)	99,91 (p=0)	7 (p<0,56)
16237	$99,46^{+-0,04}$	$7,0^{+-1,1}$	99,70 (p<0,0001)	11 (p<0,001)	99,70 (p<0,0001)	9 (p<0,07)
16238	$99,74^{+-0,02}$	$6,1^{+-0,3}$	99,89 (p<0,0001)	6 (p<0,76)	99,86 (p<0,0001)	7 (p<0,01)
16239	$99,70^{+-0,02}$	$6,2^{+-0,6}$	99,87 (p=0)	9 (p<0,0001)	99,86 (p=0)	8 (p<0,01)
global	$99,66^{+-0,21}$	$6,4^{+-1,2}$	99,84+-0,12	7,8+-1,7	99,85+-0,12	7,2+-1,1

Table 3. Résultats obtenus sur l'ensemble de test.

5 Discussion

Notre expérience avec GPS montre que l'utilisation d'une heuristique pour sélectionner les exemples à ajouter à *TS*, mène à une courbe d'apprentissage plus stable et plus proche de la courbe d'apprentissage idéale. En conséquence, la convergence est atteinte avec une meilleure exactitude avec GPS qu'avec RS. La raison en est que GPS charge au début tous les exemples les plus difficiles pour améliorer l'apprentissage. Cette méthode mène à une augmentation rapide de l'exactitude au début et à un plateau très stable à l'extrémité (les exemples additionnels sont beaucoup moins informatifs que les exemples difficiles). Ainsi plutôt que de perturber la courbe d'apprentissage [9] par une sélection aléatoire des exemples, l'addition des exemples incorrectement classés dans les itérations précédentes stabilise l'apprentissage et améliore la convergence. Cependant, la vitesse de convergence doit être étudiée plus précisément car les résultats obtenus avec les données de bradycardies ne montrent pas une différence aussi significative.

L'addition de bonnes classifications ne conduit pas à une amélioration de GPS et aurait même tendance à retarder la convergence. GPS+ pourrait par contre montrer de meilleures performances dans le cas de bases d'apprentissage très bruitées. GPS ne prenant pas en compte les exemples proches du centre de gravité, il pourrait être perturbé par des « outliers ».

La méthode proposée dans cet article permet également à l'algorithme de converger quand RS ne converge pas. Ng et Dash [11] ont également rapporté des situations dans lesquelles PS ne converge pas. Ainsi l'amélioration de PS est une nécessité pour des applications réelles.

La méthode de calcul de la distance peut être difficile à choisir. Cependant, la distance n'est pas censée être précise (elle fournit juste une évaluation de la classification) et peut être vue comme un classificateur faible. Nous partons du principe selon lequel, pour chaque problème d'apprentissage supervisé, l'utilisateur comprend suffisamment les données pour choisir une mesure de distance.

L'un des inconvénients de la distance est qu'elle ne permet pas de choisir les exemples les plus « différents » parmi les plus éloignés du centre de gravité. En effet, dans l'espace des exemples, les plus utiles sont les exemples le long des frontières. Cependant, il peut se produire des situations dans lesquelles GPS choisit des exemples difficiles dans un même secteur de l'espace des exemples. Une amélioration de la méthode pourrait consister à choisir des exemples difficiles différents pour élargir le secteur couvert par l'apprentissage dans toutes les directions (en calculant une similarité entre chaque échantillon). Cebron et Berthold [20] ont montré l'intérêt de considérer un ensemble de nouveaux exemples très diversifiés dans le cadre de l'«active learning».

Un autre avantage de GPS provient de la reproductibilité de l'apprentissage alors que des apprentissages avec RS à partir du même ensemble de données peuvent donner des résultats très différents. Comme l'un des buts est de réduire le temps d'apprentissage, il semble inopportun d'exécuter PS plusieurs fois de suite afin d'obtenir les meilleurs résultats. Avec GPS, les mêmes résultats optimaux seront toujours obtenus à partir du même ensemble de données et des paramètres.

La complexité de la mesure de distance dépend de nature de la distance choisie. Le calcul de la distance euclidienne entre chaque exemple et le centre de gravité de sa classe, par exemple, ajoute une complexité linéaire seulement à l'initialisation de la boucle. Ainsi pour de petits ensembles de données le calcul de distance peut prendre plus de temps que le reste de la boucle. Mais pour de grands ensembles de données, le temps de calcul de distance est négligeable par rapport au temps de calcul de toutes les itérations nécessaires à moins que la convergence ne soit atteinte très tôt. Cependant, une convergence précoce repose sur une mesure de distance adéquate, qu'il est difficile de déterminer a priori.

6 Conclusion

Nous avons présenté une amélioration de l'échantillonnage progressif qui utilise une sélection des exemples les plus mal classés par le modèle appris. L'échantillonnage progressif guidé (GPS) s'est avéré plus efficace que l'échantillonnage progressif aléatoire (RS) pour apprendre un classificateur à partir d'un ensemble de données volumineux. L'utilisation de la connaissance *a priori* pour guider la sélection des exemples « les plus appropriés » pour l'apprentissage mène à une meilleure convergence. GPS présente également une courbe d'apprentissage plus stable que RS. Cette approche peut également être utile dans des situations concernant de petits ensembles de données pour déterminer les meilleurs exemples.

Dans cet article, la stabilisation de la courbe d'apprentissage apportée par GPS a surtout été étudiée, de manière expérimentale. Cette stabilisation a conduit à une meilleure détection de convergence améliorant ainsi les performances du modèle appris. La contribution de cette stabilisation à l'amélioration du rapport « exactitude/temps de calcul» doit aussi être étudiée, notamment sur des bases de données plus volumineuses.

La méthode proposée permet de choisir les exemples nécessaires à un bon apprentissage. Une des perspectives d'amélioration serait de choisir les exemples nécessaires *et suffisants* par une utilisation plus judicieuse de la distance qui prendrait en compte la dispersion des exemples candidats dans l'espace d'apprentissage. L'ensemble d'apprentissage serait ainsi composé d'exemples non redondants et serait donc le plus petit possible. Enfin, le choix d'une bonne distance est difficile et l'étude des distances adaptées à des données non numériques doit être entreprise.

Bibliographie

[1] Chawla N, Eschrich S et Hall LO, Creating Ensembles of Classifiers, Dans *First IEEE International Conference on Data Mining*, 2001.

[2] Chawla NV, Moore TE, Hall LO, Bowyer KW, Kegelmeyer WP et Springer C, Distributed learning with bagging-like performance, *Pattern Recognition Letters*, **24**, pp. 455-471, 2003.

[3] Breiman L, Bagging predictors, *Machine Learning*, **24**, pp. 123-140, 1996.

[4] Schapire R, Rochery M, Rahim M, Gupta N, Incorporating prior knowledge into boosting, Dans *Nineteenth International Conference on Machine Learning*, 2002.

[5] Macek J, Incremental Learning of Ensemble Classifiers on ECG Data, Dans *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, 2005.

[6] Quinlan JR. Learning Efficient Classification Procedures and Their Application to Chess and Games. In Michalski RS, Carbonell JG, Mitchell TM (Eds.), *Machine Learning. An Artificial Intelligence Approach*. 1983.

[7] Fürnkranz J, Integrative Windowing, *Journal of Artificial Intelligence Research*, **8**, pp. 129-164, 1998.

[8] Maloof MA et Michalski RS, Incremental learning with partial instance memory, *Artificial Intelligence*, **154(1-2)**, pp. 95 - 126, 2004.

[9] Provost F, Jensen D, Oates T, Efficient progressive sampling, Dans *fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999.

[10] Breiman L, Pasting bites together for prediction in large data sets, *Machine Learning*, **36**, pp. 85-103, 1999.

[11] Ng W, Dash M, An Evaluation of Progressive Sampling for Imbalanced Data Sets, Dans *6th IEEE International Conference on Data Mining Workshops (ICDM 2006)*, 2006.

[12] Estrada A, Morales EF, NSC: A New Progressive Sampling Algorithm, Dans *workshop on Machine Learning for Scienific data Analysis (IBERAMIA 2004)*, 2004.

[13] Leite R et Brazdil P, Improving Progressive Sampling via Meta-learning, Dans *Progress in Artificial Intelligence*, 2003.

[14] Gu B, Liu B, Hu F et Liu H, Efficiently Determining the Starting Sample Size for Progressive Sampling, Dans *Proceedings of the 12th European Conference on Machine Learning*, 2001.

[15] Parthasarathy S, Efficient Progressive Sampling for Association Rules, Dans *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, 2002.

[16] Chuang K, Chen M, Yang W, Progressive Sampling for Association Rules Based on Sampling Error Estimation, Dans *Advances in Knowledge Discovery and Data Mining*, 2005.

[17] Portet F, Gao F, Hunter J, Quiniou R, Reduction of Large Training Set by Guided Progressive Sampling: Application to Neonatal Intensive Care Data, Dans *Intelligent Data Analysis in bioMedicine and Pharmacology (IDAMAP-2007)*, 2007.

[18] John GH et Langley P, Static Versus Dynamic Sampling for Data Mining, Dans *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996.

[19] Stamatopoulos C, Observations on the geometrical properties growth in sampling with finite populations, FAO fisheries technical paper, 1999.

[20] Cebron N, Berthold MR, Adaptive Active Classification of Cell Assay Images, Dans *10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2006)*, 2006.

[21] Provost F et Kolluri V, A Survey of Methods for Scaling Up Inductive Algorithms, *Data Mining and Knowledge Discovery* , **3(2)**, pp. 131 - 169, 1999.

[22] Newman D, Hettich S, Blake C, Merz et C, *UCI repository of machine learning databases*, http://www.ics.uci.edu/~mlearn/MLRepository.html [Access: 2007], 1998.

[23] Quinn JA, Williams CKI, Known Unknowns: Novelty Detection in Condition Monitoring, Dans *Pattern Recognition and Image Analysis*, 2007.